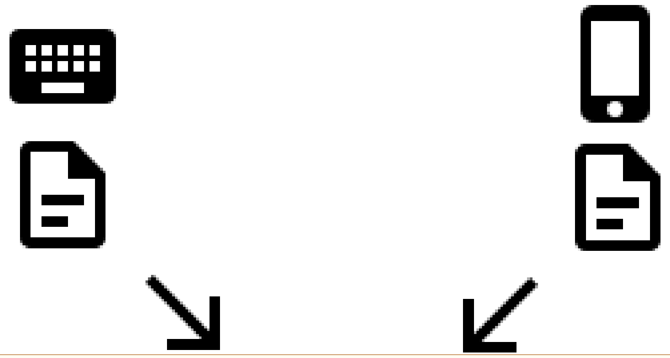
INPUT MULTIPLEXER

INTRODUCING THE INPUTMULTIPLEXER

The Next Step On InputProcessors

- What if we want to use multiple InputProcessors?
- We may want an InputProcessor for the keyboard and a GestureDetector for an Android touch screen. Maybe a second InputProcessor for a player two. Or just for organizing the code.
- This doesn’t work:



```
Gdx.input.setInputProcessor(keyboardInput);
Gdx.input.setInputProcessor(GestureInput);
```

INTRODUCING THE INPUTMULTIPLEXER II

Fixing The Problem

- We fix this problem with the InputMultiplexer.
- First we create the InputMultiplexer.

```
InputMultiplexer multiplexer = new InputMultiplexer();
```

- Next we add the InputProcessors.

```
multiplexer.addProcessor(KeyboardInput);
multiplexer.addProcessor(GestureInput);
```

- Finally, we set the multiplexer as the InputProcessor.

```
Gdx.input.setInputProcessor(multiplexer);
```

INPUTPROCESSOR RETURN

```
@Override
public boolean keyDown(int keycode) {
     return false;
}
```

- The events return a boolean so you can decide if you want to process the next event of the same type, or not, once you have set multiple InputProcessors.
- If it returns true it stops, otherwise it goes to the same event type of the next InputProcessor.